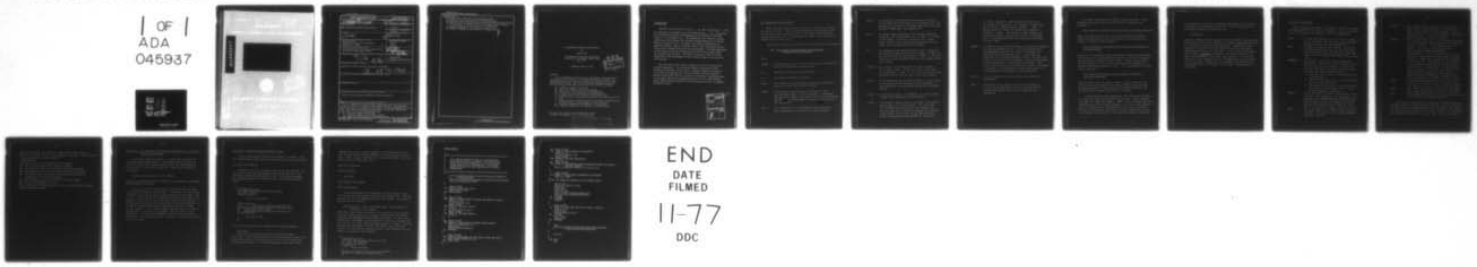
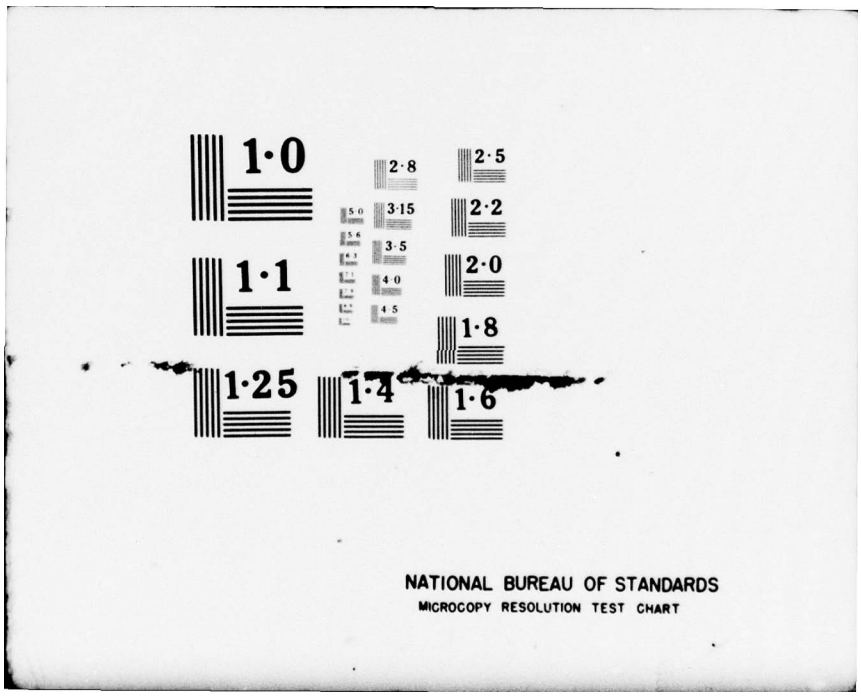
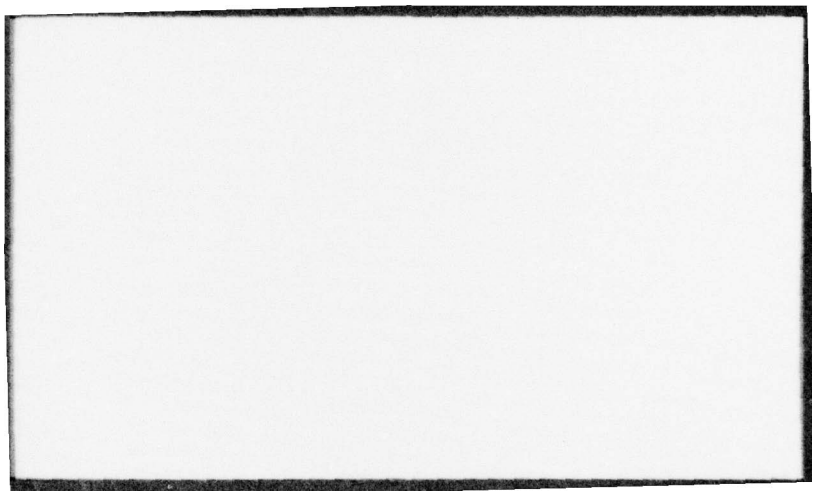
AD-A045 937 RICE UNIV HOUSTON TEX DEPT OF ELECTRICAL ENGINEERING F/G 9/5
AN INTERACTIVE CLUSTER DISPLAY SYSTEM.(U)
MAY 77 C PAU
UNCLASSIFIED EE-TR-7709 AFOSR-75-2777 AFOSR-TR-77-1246 NL

1 OF 1
ADA
045937



END
DATE
FILMED
11-77
DDC


1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5
5·0
5·6
6·3
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (19) (18) AFOSR-TR--77-1246	2. GOVT ACCESSION NO. (9) 3. RECIPIENT'S CATALOG NUMBER Technical rept.
4. TITLE (and Subtitle) (6) AN INTERACTIVE CLUSTER DISPLAY SYSTEM.	5. TYPE OF REPORT & PERIOD COVERED Interim
	6. PERFORMING ORG. REPORT NUMBER EE 7709
7. AUTHOR(s) (10) Charles Pau	8. CONTRACT OR GRANT NUMBER(s) (15) AFOSR-75-2777
9. PERFORMING ORGANIZATION NAME AND ADDRESS Rice University Department of Electrical Engineering Houston, TX. 7700;	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 2304/A2
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research (N/M) Bolling AFB, Washington DC 20332	12. REPORT DATE (11) May, 1977 13. NUMBER OF PAGES (12) 17p.
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) (16) 2304 (17) A2	15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

(14) EE-TR-7709

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Cluster displays; pattern recognition; feature extraction

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A set of FORTRAN subroutines has been implemented at Rice University for the display of clusters on a Tektronix 4013 graphics terminal. The subroutines display each point as a character on a portion of the screen determined by the user. Once the main subroutine has been called, the user can interactively select one or more of the following functions:

(1) Displays a sample of the points;
(2) Selectively displays one or more clusters;
(3) Zooms into a small section of the display area;

(OVER)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED 403244

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (Continued) --

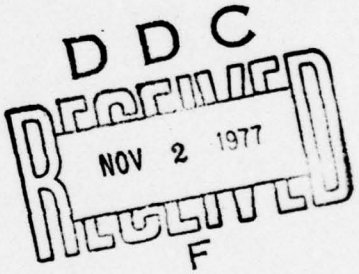
(4) Displays only a small section of the display area;
(5) Displays all but a small section of the display area;
(6) Constructs decision boundaries according to criteria set up by the user;
(7) Allows the user to identify points in the display area and returns the coordinates in an array upon exit from subroutine;
(8) Produces a hardcopy of the display on a line printer;
(9) Produces a hardcopy of the display on a CalComp plotter.

AN INTERACTIVE CLUSTER DISPLAY SYSTEM

by

Charles Pau

Department of Electrical Engineering
Rice University, Houston, Texas 77001
May, 1977



TECHNICAL REPORT EE 7709

ABSTRACT:

A set of FORTRAN subroutines has been implemented at Rice University for the display of clusters on a Tektronix 4013 graphics terminal. The subroutines display each point as a character on a portion of the screen determined by the user. Once the main subroutine has been called, the user can interactively select one or more of the following functions:

(1) Displays a sample of the points;
(2) Selectively displays one or more clusters;
(3) Zooms into a small section of the display area;
(4) Displays only a small section of the display area;
(5) Displays all but a small section of the display area;
(6) Constructs decision boundaries according to criteria set up by the user;
(7) Allows the user to identify points in the display area and returns the coordinates in an array upon exit from subroutine;
(8) Produces a hardcopy of the display on a line printer;
(9) Produces a hardcopy of the display on a CalComp plotter.

This work was supported by the AFOSR Grant 75-2777
Principal Investigator: Rui J. P. de Figueiredo

INTRODUCTION

Consider a set of points P: $x_1, x_2, \ldots x_n$ where $x_i=(x_{i1}, x_{i2}, \ldots x_{im})$ and each point x_i belongs to one of k classes $C_1, C_2, \ldots C_k$. The process of classification involves the designing of the function f such that $f(x_i) = j$ if x_i belongs to C_j. The designing of the function is often simplified if the designer can visulize the distribution of the points. When m is equal to 1 or 2, the points can simply be displayed on a graphics terminal. When m is greater than 2, only a projection of the points onto a 2-dimensional plane can be displayed. The system described in this report can be used to display the points before and/or after the classifi ation process. The various interactive features of the system also allow the designer to have a better grasp of the structure of the points.

The first part of this report contains a list of the subroutines and their parameters. The second part gives a description of the principle of operation of the system. Appendix A gives a description of the Tektronix 4013 graphics terminal and the Tektronix Terminal Control System (TCS) that are used by the subroutines. Appendix B gives a list of the Job Control Language (JCL) and a list of Time Sharing Option (TSO) commands needed for link-editing and execution of programs using this system.

THE SUBROUTINES IN THE SYSTEM

There are four subroutines in the system. The user can call any one of the four. Under normal operation, the only one called by the user is the subroutine CLUST. The other subroutines are called by the user only if he/she wants to bypass the interactive capabilities of the system.

The subroutine CLUST is invoked by the subroutine call statement:

```
CALL CLUST(LEGEND,N,M,MPTR,MFLAG,PTS,IORIG,SNS,SCALES,
XY,CUR,ICFLAG,PFLAG,ICHAR,IPTS)
```

where:

LEGEND --- An 18 elements array of 4 characters each to be printed under the display for identification.

N --- The number of points to be displayed.

M --- The number of different classes.

MPTR --- An M elements array of pointers pointing to the first element of each class in the list PTS.

MFLAG --- A 2M elements logical array to indicate if the corresponding class is to be displayed. If MFLAG(J) has the value .FALSE., the class J will not be displayed. The (M+1)th to 2Mth elements of the array are used in the subroutine.

PTS --- A 2 x N elements array (PTS(2,N)) containing the x- and y-coordinates of the points to be displayed.

IORIG --- A 2 elements array containing the screen coordinates of the lower-left-hand corner of the display area. The display area will be the rectangle defined by IORIG and the point (1000,750). (See Appendix A)

SNS --- An integer flag indicating if the display should be scaled to cover the display area. If SNS = 1, scaling is performed. If SNS = 0, the limits (i.e. the x- and y- values at the lower-left-hand corner) and the scales will be supplied by XY and SCALES.

XY --- A 2 elements array containing the x- and y- values of the lower-left-hand corner of the display. If SNS = 0, this will be used in the display. If SNS = 1, this will return the values obtained from the scaling used in the last display.

SCALES --- A 2 elements array containing the scales used for the x- and y- axes. If SNS = 0, this will be used in the display. If SNS = 1, this will return the values obtained from the scaling used in the last display.

CUR --- A 2 x 10 elements array containing the x- and y- values of the points on the piecewise linear boundaries if such boundaries are constructed during the interactive session.

ICFLAG --- A 10 elements array. If ICFLAG(J) = 0, the Jth point in CUR is the beginning of a new boundary.

PFLAG --- A 2N elements logical array to indicate if the corresponding point is to be displayed. This gives the user an additional control over the display (e.g. the user might want to display randomly only one-half of the points in one class, he can achieve this by simply setting half of the PFLAG's for that class

to .FALSE. randomly). PFLAG can be altered during the session by the system. For example, if the system returns after a ZOOM operation, PFLAG(J) = .TRUE. if the Jth point is inside the zoomed area, otherwise PFLAG(J) = .FALSE.. The (N+1)th to 2Nth elements are used by the system. Notice that a point is displayed only if the PFLAG and MFLAG corresponding to the point are both .TRUE..

ICHAR --- A M elements array each containing a 4 characters mnemonic for the corresponding class. The last character must be alphanumeric and will be used in the display to represent the corresponding class. If the last two characters are the same alphabetic letter, the lower case letter will be used in the display. If the last character is not alphanumeric, an asterisk will be used. Thus there are a possible of 63 classes (52 letters + 10 numbers + asterisk). In the hardcopy, a lower case letter will be represented by the letter with an underscore.

IPTS --- A 2 x N elements array to be used by the system for hardcopying.

NCH --- The logical unit number of the file for hardcopies. If this is not defined by the user, the user will be queried when he/she tries to get a hardcopy.(See Appendix B)

The other subroutines are CLUST1, CLUST2, and FIND. CLUST1 displays the clusters on the terminal and is invoked by the statement:

```
CALL CLUST1(PTS,ICHAR,N,M,MPTR,PFLAG,MFLAG,IORIG,SNS,SCALES,XY)
```

where the parameters are the same as those defined for CLUST. CLUST2 put the hardcopy into a file which can be printed later. CLUST2 is invoked by the statement:

```
CALL CLUST2(PTS,IC,N,M,MPTR,PFLAG,MFLAG,IORIG,SNS,SCALES,XY,
IPTS,NCH,LEGEND)
```

where IC is a M elements array containing the ASCII representation of the characters representing the classes. The other parameters are the same as those defined for CLUST. FIND gives the values of XY and SCALES if they are not defined and also compute the maximum and minimum values of x and y. This subroutine is invoked by the statement:

```
CALL FIND(PTS,N,M,PFLAG,MFLAG,SNS,SCALES,XY,MPTR,LX,LY,
XMIN,XMAX,YMIN,YMAX)
```

where LX and LY are the number of points on the screen for the x- and y- axes (See Appendix B), XMIN and XMAX are the minimum and maximum values of the x interval and YMIN and YMAX are the minimum and the maximum values of the y interval.

In additional to these subroutines, the system also contains four functions which may be useful to the user. These functions are ICODE, LE, ICONV and ICONV2. ICODE is used to decode the user's command during the interactive session. LE is a trivial function and is included because IBM FORTRAN does not allow a character string in an assignment statement but allows a character string as an argument of a function. The function LE makes it possible

to use character strings in assignment and logical IF statements. For example, to assign the string 'this' to the variable K, the following statement will do the trick:

```
K=LE('THIS')
```

ICONV and ICONV2 are used to convert from the EBCDIC representation of characters used in the IBM 370 to the ASCII representation used by the Tektronix terminal, and vice versa. ICONV takes a four characters character string (in EBCDIC) and returns the ASCII representation of the last character. If the last two characters are the same alphabetic character, the value returned is the ASCII representation of the lower case character. If the last character is not alphanumeric, the representation of an asterisk will be returned. ICONV2 returns the EBCDIC representation of the argument which is assumed to be the ASCII representation of a character.

Principle of Operation

When the subroutine CLUST is invoked, it will first display the data passed to it through the arguments. The user can then input one or more of ten commands. The ten commands are:

PRINT --- To dump the display on a print file. In order to do so, a dataset must first be created and then allocated to the file (See Appendix B).

ZOOM --- To zoom in on part of the display. The cursor will be activated for the user to specify the lower left hand corner and the upper right hand corner of the rectangle that he/she wants to zoom into.

BOUNDARY --- To construct a piecewise linear boundary or boundaries. The cursor will be invoked for the user to construct the boundary. The user can position the cursor to a point of discontinuity and then type one of three characters:

'm' indicates that the point is the beginning of a new boundary,

'd' indicates that the point is a continuation of the current boundary, and

'e' indicates that the point is a continuation of the current boundary and is the last point of the boundary or boundaries.

NOSCALE --- To set SNS to 0 to indicate that scaling is not required. For example, if this command is used together with the ZOOM command, the display will show the points within the zoomed area but the size of the zoomed area will remain the same. SNS is set to 0 until a SCALE command is issued.

SCALE --- To set the SNS to 1 to indicate that scaling is required. SNS remains 1 until the next NOSCALE command is issued.

CHOOSE --- This command allows the user to select one or more classes to be displayed. The system will list all the classes by their mnemonic. The user can then put a '1' next to the classes that he/she wants to include in the next display.

INVERT --- This command allows the user to display points that are not previously displayed because their PFLAG's are set as .FALSE.. This command inverts the PFLAG's. The user is asked the question: "DO YOU WANT TO INVERT EVERYTHING?" If the answer is 'YES', all the PFLAG's will be inverted. Otherwise only those PFLAG's that has a value of .TRUE. when the subroutine is entered will be inverted. For example, if this command is issued after a ZOOM, the points outside the zoomed area will be displayed. Notice that this does not change the values of the MFLAG's. So a point which is previously suppressed by its MFLAG will remain suppressed. The only way to change the MFLAG's is through the CHOOSE command.

RESET --- A copy of MFLAG and PFLAG at the time that the subroutine is entered is kept. The RESET command will copy this into the current MFLAG and PFLAG.

HELP --- This command will cause a list of all the commands and a short description of each to be displayed.

END --- This command will terminate the subroutine and return control back to the calling program.

The commands are entered as a sequence of commands separated by commas and has no blank except at the end. The first letter of each command is sufficient to define the command. The commands are executed in order from left to right. There is no limitation to the number of commands in a line. After a line of commands

have been executed, the system is ready for the next sequence (unless, of course, if one of the commands is an END). The following is an example of a commands sequence:

C,ZOOM,PR,I,P,R

which will cause the following things to happen:

(1) The user can choose the classes to display,

(2) The user can zoom in to a certain area of the display,

(3) The zoomed area is put into a file for hardcopy output,

(4) The PFLAG's are inverted so that the points outside the zoomed area will now be displayed,

(5) The display is put into a file for hardcopy output,

(6) The MFLAG and PFLAG are reset.

Notice that a new display appears on the screen after every command except the PRINT command.

Appendix A: The Tektronix 4013 Graphics Terminal and the Terminal Control System (TCS)

The Tektronix 4013 terminal is a storage CRT terminal with 1024 x 781 addressable points. The Terminal Control System is a set of routines which may be used to draw lines and points on the terminal, to erase the screen, etc. Users are referred to the manual:

Terminal Control System - Users Manual

which may be obtained from the bookstore or referenced in the Users Clinic and Rm. 6 in ICSA.

The main feature of the TCS which is of particular interest to the users of the CDS is the cursor. The cursor is a set of two crosshairs which can be used to specify points on the screen. The cursor is controlled by two thumbwheels located to the right of the keyboard. After the user has positioned the cursor, he/she can hit any one of the keys and the position of the cursor together with the ASCII representation of the character struck will be sent to the computer. In the case of ZOOM, where the cursor is used to specify the two corners of a rectangle, the key used in inconsequential. In the case of BOUNDARY, the key used specified if the point is the starting point of a boundary, a continuation point, or the last point.

Appendix B: Using the Cluster Display System

A basic knowledge of JCL and TSO commands is assumed. Users not familiar with either one are referred to the respective manuals.

Setting up the program

In order to set up a program to be used on the terminal, the user needs to link his/her program with the CDS and TCS. This can be done either through batch or TSO. The following example will demonstrate the JCL required to set up the program through batch:

```
cc 1
   /*PASSWORD password
   //jobname JOB (Standard JOB card format)
   //   EXEC  FTG1CL
   //FORT.SYSIN DD *

           user's source deck

   /*
   //LKED.SYSLIB DD
   //             DD DSN=RICE.GRAPHICS.LOAD,DISP=SHR
   //LKED.SYSLMOD DD DSN=userid.name.LOAD(membername),
   //    DISP=(,CATLG),UNIT=SYSDA,
   //    SPACE=(TRK,(2,1,1),RLSE),DCB=BLKSIZE=13030
   //LKED.SYSIN DD *

           CDS object deck
   /*
   //
```

The same setup can be done on TSO by the following commands:

```
FORT name1
LINK (name1 name2) FORTLIB LIB('RICE.GRAPHICS.LOAD)
```

where name1 is the dataset name of the dataset conatining the user's program and name2 is the dataset name of the dataset containing the CDS object module. If another language is used other than

FORTRAN (e.g. PL/I), proper linkage to the FORTRAN subroutines must be maintained and SYS1.FORTLIB is included in the link edit step. Other library subroutines can also be included in the link edit step (e.g. IMSL, SSP, etc.).

Running the Program

The TSO command:

```
CALL name
```

will execute the program.

Getting Hardcopies

To get hardcopies of the displays, the output must be put in a dataset and the dataset printed on the lineprinter. The dataset must be allocated before executing the program. Allocation is done by the TSO command:

```
ALLOC DA(dataset name) F(FTxxF001) [NEW  TRACKS SPACE(1,1)
      USING(attr-list-name)]
```

where the items inside the brackets are used only if the dataset is to be created and "attr-list-name" is defined by the ATTRIB command to be RECFM(FB) BLKSIZE(1330) LRECL(133). If the channel number (xx in FTxxF001) is not specified in the user program, CDS will ask for the number when the PRINT command is issued. The user should enter the number using an (I2) format. Note that the terminal is allocated to channels 5 and 6. The following JCL will produce a hardcopy of the displays:

```
cc 1
   /*PASSWORD password
   //jobname JOB (Standard JOB card format)
   //   EXEC  PGM=IEBPTPCH
   //SYSPRINT DD SYSOUT=A
   //SYSIN DD *
            PRINT PREFORM=A
   /*
   //SYSUT1 DD DSN=userid.dataset-name,DISP=OLD
   //SYSUT2 DD SYSOUT=A,DCB=BLKSIZE=133
   //
```

SAMPLE PROGRAM

```
C
C*****************************************************************
C
C     THIS PROGRAM DEMONSTRATES HOW THE CLUSTER DISPLAY
C     SYSTEM CAN BE USED IN A PROGRAM.  THE PROGRAM READS
C     IN ALL THE NECESSARY PARAMETERS TO TAKE THE LINEAR
C     COMBINATIONS OF UP TO 20 DIMENSIONS.  THE PROGRAM
C     ALLOWS THE DISPLAY OF MORE THAN ONE SETS OF LINEAR
C     COMBINATIONS.
C
C*****************************************************************
C
      INTEGER LEGEND(18),MPTR(20),IORIG(2),ICFLAG(10),ICHAR(20),
     &        IPTS(2,2000),SNS
      REAL PTS(2,1000),SCALES(2),CUR(2,10),VAL(20),WTS(2,20),XY(2)
      LOGICAL MFLAG(40),PFLAG(2000)
C
C
      WRITE (6,901)
 901  FORMAT(' ENTER TITLE LINE')
      READ (5,801) LEGEND
 801  FORMAT(18A4)
C
C
 10   WRITE (6,902)
 902  FORMAT (' ENTER NUMBER OF CLASSES AND NUMBER OF POINTS')
      READ (5,*) M,N
      IF (M.LE.20) GO TO 20
      WRITE (6,903)
 903  FORMAT (' TOO MANY CLASSES')
      GO TO 10
 20   IF (N.LE.1000) GO TO 30
      WRITE (6,904)
 904  FORMAT (' TOO MANY POINTS')
      GO TO 10
C
C
 30   WRITE (6,905)
 905  FORMAT (' ENTER NUMBER OF POINTS IN EACH CLASS')
      READ (5,*) (MPTR(I),I=1,M)
      DO 40 I=2,M
      MPTR(I)=MPTR(I)+MPTR(I-1)
 40   CONTINUE
C
C
      WRITE (6,906)
 906  FORMAT (' ENTER NAME FOR EACH CLASS (1 NAME PER LINE)')
      READ (5,802) (ICHAR(I),I=1,M)
 802  FORMAT (A4)
C
C
```

```
 50   WRITE (6,907)
 907  FORMAT (' ENTER NUMBER OF DIMENSIONS')
      READ (5,*) ID
      IF (ID.LE.20) GO TO 60
      WRITE (6,908)
 908  FORMAT (' TOO MANY DIMENSIONS')
      GO TO 50
 60   WRITE (6,909)
 909  FORMAT (' ENTER WEIGHT FOR EACH DIMENSION (FIRST THE X-AXIS,',
     &        ' THEN THE Y-AXIS)')
      READ (5,*,END=999) ((WTS(I,J),J=1,ID),I=1,2)
C
C
      WRITE (6,910)
 910  FORMAT (' ENTER SCREEN COORDINATES FOR ORIGIN')
      READ (5,*) IORIG
C
C***** THE POINTS ARE ASSUMED TO BE IN LOGICAL UNIT 4
C
      DO 70 I=1,N
      READ (4,*) (VAL(J),J=1,ID)
      PTS(1,I)=0.
      PTS(2,I)=0.
      DO 71 II=1,ID
      PTS(1,I)=PTS(1,I)+VAL(II)*WTS(1,II)
      PTS(2,I)=PTS(2,I)+VAL(II)*WTS(2,II)
 71   CONTINUE
 70   CONTINUE
      REWIND 4
C
C
      WRITE (6,911)
 911  FORMAT (' DISPLAYING ONE EVERY N POINTS, ENTER N')
      READ (5,*) KT
      DO 80 I=1,N
      PFLAG(I)=MOD(I,KT).EQ.0
 80   CONTINUE
      DO 90 I=1,M
      MFLAG=.TRUE.
 90   CONTINUE
C
C
      SNS=1
      CALL CLUST(LEGEND,N,M,MPTR,MFLAG,PTS,IORIG,SNS,SCALES,
     &           XY,CUR,ICFLAG,PFLAG,ICHAR,IPTS)
C
C
      GO TO 60
C
C
 999  STOP
      END
```